



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|----------------------|---------------------|------------------|
| 10/799,463 | 03/12/2004 | Dennis W. Minium JR. | 13768.1293 | 7693 |
| 47973 7590 01/11/2010 WORKMAN NYDEGGER/MICROSOFT 1000 EAGLE GATE TOWER 60 EAST SOUTH TEMPLE SALT LAKE CITY, UT 84111 | | | | |
| EXAMINER | | | | |
| ZHEN, L F B | | | | |
| ART UNIT | | PAPER NUMBER | | |
| 2194 | | | | |
| MAIL DATE | | DELIVERY MODE | | |
| 01/11/2010 | | PAPER | | |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/799,463

Applicant(s)

MINIUM ET AL.

Examiner

LI B. ZHEN

Art Unit

2194

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 17 September 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13, 17-21, 23-33 and 38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-13, 17-21, 23-33 and 38 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-506)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Claims 1 – 13, 17 – 21, 23 – 33 and 38 are pending in the application.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/17/2009 has been entered.

Response to Arguments

3. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. **Claims 23 – 33 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,174,348 to Sadhu et al. [hereinafter Sadhu,**

previously cited] in view of U.S. Application Publication No. 20030046282 to Carlson et al. [hereinafter Carlson].

6. As to claim 23, Sadhu teaches a computer-readable storage medium having computer-executable instructions for performing a method for facilitating an interface between non-integrated applications [col. 3, lines 26 – 32], the method comprising:

providing a first application program interface (API) [col. 3, lines 55 – 65], the first API configured as an artifact provider [submodule 31c-1; col. 7, lines 4 – 24];

exposing a referenced artifact hosted by the first application via the artifact provider [submodule 31c-1; col. 7, lines 4 – 24];

providing a second application program interface (API) [col. 5, lines 25 – 31 of Sadhu], the second API configured as an artifact consumer that communicates with a second non-integrated application [collaboration module; col. 5, lines 35 – 49];

exposing a reference held by second application and the referring artifact associated [col. 15, lines 21 – 36] with the reference via the artifact consumer [col. 5, lines 35 – 49]; and

linking the referring artifact to the referenced artifact via the reference, the reference including an artifact identifier [col. 17, line 35 – col. 18, line 20] of the referenced artifact [col. 5, lines 32 – 48; col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57]. Sadhu does not specifically disclose an artifact provider that communicates with a first non-integrated application, the artifacts of the first application including an item of public data of the first application, the second application including

a referring artifact comprising an item of public data of the second application, the second application being configured to access the items of data of the artifacts of the first application via the first API, the first application being configured to access the references of the artifacts of the second application via the second API.

However, Carlson teaches an artifact provider that communicates with a first non-integrated application [external access; paragraph 0093], a first application program interface (API) of a first application, the first API configured as an artifact provider [Asset source 12A manages artifact storage 60 to store artifact data retrieved from repository 4A as needed, and provides artifact interface 53 for external access; paragraphs 0093 and 0072], a second application program interface (API) of a second application, the second API configured as an artifact consumer [capture logic 112; paragraphs 0111 – 0112], the artifacts of the first application including an item of public data of the first application [provides artifact interface 53 for external access; paragraph 0093], the second application including a referring artifact comprising an item of public data of the second application [asset capture tool next identifies any custom data types used by the external functions of the asset; paragraph 0130], the second application being configured to access the items of data of the artifacts of the first application via the first API [asset source 12A may include a writable interface 52 that allows asset capture module 26 to augment the artifact information of the underlying repository 4A or artifact storage 60; paragraph 0095], the first application being configured to access the references of the artifacts of the second application via the second API [asset capture tool next identifies any custom data types used by the external functions of the asset,

and allows the user to select any additional data types that are part of the API for the asset; paragraph 0130].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention of Sadhu to include the features of Carlson. One of ordinary skill in the art would have been motivated to make the combination because this provides generalized interfaces to diverse repositories, and to generate software assets in a normalized form that complies with a data description language [paragraph 0014 of Carlson].

7. As to claim 38, Sadhu as modified teaches a computer-implemented system that facilitates data integration among one or more non-integrated applications in a development environment [col. 3, lines 26 – 32 of Sadhu], comprising:

at least one processor, coupled to a memory, that executes the following computer-executable components [col. 3, lines 40 – 48 of Sadhu]:

an integration service in the development environment [software development tool integration module 31c, Fig. 2; col. 5, lines 20 – 31 of Sadhu] that includes one or more non-integrated applications [collaboration module; col. 5, lines 35 – 49 of Sadhu] that each comprise at least one artifact [col. 2, lines 59 – 61 and col. 7, lines 4 – 24 of Sadhu], the integration service comprises:

a first application [col. 5, lines 20 – 31 of Sadhu] and a second application [col. 17, lines 35 – 49 of Sadhu] that each include one or more artifacts [col. 7, lines 4 – 24 and col. 6, lines 6 – 23 of Sadhu], the one or more artifacts are items of data of the

applications that are publicly exposed [paragraph 0093 of Carlson], the one or more artifacts include artifact types and unique artifact identifiers [paragraph 0132 of Carlson];

a first application program interface (API) configured as an artifact provider [col. 7, lines 4 – 24 of Sadhu and paragraphs 0093 and 0072 of Carlson] associated with the first application, the artifact provider that facilitates exposing at least a referenced artifact [col. 7, lines 3 – 24 of Sadhu] of the first application [paragraph 0093 of Carlson];

a second application program interface (API) configured as an artifact consumer [col. 5, lines 35 – 49 of Sadhu and paragraphs 0111 – 0112 of Carlson] associated with the second application, the artifact consumer that facilitates exposing at least a referring artifact of the second application [col. 15, lines 21 – 36 of Sadhu] and a reference associated with the referring artifact [col. 17, line 35 – col. 18, line 20 of Sadhu], the reference includes an artifact identifier corresponding to the referring artifact exposed by the artifact provider [paragraph 0130 of Carlson], the second application being configured to access the referenced artifact of the first application via the first API [paragraph 0095 of Carlson], the first application being configured to access the referring artifact of the second application via the second API [paragraph 0130 of Carlson]; and

a linking component that facilitates creation of a link between the referring artifact and the referenced artifact via the reference included in the artifact consumer [col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57 of Sadhu], the link includes a link type that indicates a type of relationship between the referring artifact and the

referenced artifact [col. 5, lines 32 – 48; col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57 of Sadhu].

8. As to claim 24, Sadhu teaches registering an artifact type for the referring artifact and the referenced artifact; and registering a link type that the referring artifact and the reference artifact hosts [col. 18, lines 15 – 20].

9. As to claim 25, Sadhu teaches presenting dependency information of the referenced artifact to a user, the information including at least one of link type [col. 14, lines 15 of Sadhu], artifact type [col. 5, lines 32 – 48; col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57 of Sadhu], artifact name [col. 14, lines 5 – 17 of Sadhu], and modification date [col. 15, lines 50 – 55 of Sadhu].

10. As to claim 26, Sadhu as modified teaches at least one of the artifact consumer or artifact provider is a web service [paragraph 0061 of Carlson].

11. As to claim 27, Sadhu as modified teaches generating an artifact proxy that represents data stored in a non-integrated application [paragraph 0060 of Carlson].

12. As to claim 28, Sadhu teaches the referenced artifact and referring artifact are representative of at least one of a source file [a fifth artifact which is the source code; col. 2, lines 59 – 61 and col. 7, lines 4 – 24], defect, requirement [functional

specifications; col. 7, lines 4 – 24 and col. 6, lines 6 – 23], test result [test documents; col. 7, lines 4 – 24], or build.

13. As to claim 29, Sadhu as modified teaches linking comprises creating a link between the referring artifact and the referenced artifact that includes a referring URI [paragraph 0132 of Carlson], a referenced URI [paragraph 0118 of Carlson], and a link type [col. 5, lines 32 – 48; col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57 of Sadhu].

14. As to claim 30, Sadhu teaches discovering which referring artifacts hold links to a specific referenced artifact [col. 11, lines 21 – 31].

15. As to claim 31, Sadhu teaches raising an event when the referenced artifact is at least one of created, deleted, and changed [col. 5, lines 59 – 67].

16. As to claim 32, Sadhu as modified teaches providing external addressability for the referenced artifact by the artifact provider [paragraph 0093 of Carlson].

17. As to claim 33, Sadhu as modified teaches wherein the first application is a source control application [paragraph 0060 of Carlson]; and wherein the second application is a defect tracking application [paragraphs 0063 – 0064 of Carlson].

18. **Claims 1 – 13, 17 – 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,174,348 to Sadhu et al. [hereinafter Sadhu, previously cited] in view of U.S. Application Publication No. 20030046282 to Carlson et al. [hereinafter Carlson] and further in view of “Modeling Software Defect Introduction” to [Chulani].**

19. As to claim 1, Sadhu teaches a system that facilitates the interface of non-integrated applications [col. 3, lines 26 – 32], comprising:

a processor coupled to memory that retains [col. 3, lines 40 – 48]:

a first application program interface (API) of a first application [software development tool integration module which has one interface that interacts with the object module; col. 3, lines 55 – 65], the first API configured as an artifact provider [submodule 31c-1; col. 7, lines 4 – 24] that hosts artifacts of the first application [software development tool integration module 31c, Fig. 2; col. 5, lines 20 – 31], each artifact of the first application being associated with an artifact type, the artifact type comprising a source file [a fifth artifact which is the source code; col. 2, lines 59 – 61 and col. 7, lines 4 – 24], a requirement [functional specifications; col. 7, lines 4 – 24 and col. 6, lines 6 – 23], or a test result [test documents; col. 7, lines 4 – 24]; and

a second application program interface (API) of a second application [modules 31a-31d interact with the object module 31e through internal interfaces 31g-31j respectively; col. 5, lines 25 – 31], the second API configured as an artifact consumer [collaboration module 31d; col. 5, lines 35 – 49] that hosts artifacts [PERSON object

41d; col. 15, lines 21 – 36] of the second application [col. 17, lines 35 – 49], each reference [col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57] being associated with one referring artifact hosted by the second application and is a link to one referenced artifact of the first application hosted by the artifact provider [submodule 31d-1 insures that nobody takes part in the review unless they are represented by one of the PERSON objects 41d which are linked to one of the REVIEWER objects 41t in the database 32; col. 17, line 35 – col. 18, line 20], the link further comprises a link type that describes a relationship between the referring artifact of the second application and the referenced artifact of the first application [PROCESS object which is linked by a plurality of relationships to many other different types of objects; col. 5, lines 32 – 48; col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57]. Sadhu does not specifically disclose the artifact type comprising a defect or version, the artifacts of the first application including items of data the first application publicly exposes to other applications, the artifacts of the second application including items of data the second application publicly exposes to other applications, and the second application being configured to access the items of data of the artifacts of the first application via the first API, the first application being configured to access the references of the artifacts of the second application via the second API.

However, Carlson teaches an artifact type comprising a build [version; paragraph 0101], a first application program interface (API) of a first application, the first API configured as an artifact provider [Asset source 12A manages artifact storage 60 to store artifact data retrieved from repository 4A as needed, and provides artifact interface

53 for external access; paragraphs 0093 and 0072], a second application program interface (API) of a second application, the second API configured as an artifact consumer [capture logic 112; paragraphs 0111 – 0112], the artifacts of the first application including items of data the first application publicly exposes to other applications [provides artifact interface 53 for external access; paragraph 0093], the artifacts of the second application including items of data the second application publicly exposes to other applications [asset capture tool next identifies any custom data types used by the external functions of the asset; paragraph 0130], and the second application being configured to access the items of data of the artifacts of the first application via the first API [asset source 12A may include a writable interface 52 that allows asset capture module 26 to augment the artifact information of the underlying repository 4A or artifact storage 60; paragraph 0095], the first application being configured to access the references [paragraphs 0117 – 0118] of the artifacts of the second application via the second API [asset capture tool next identifies any custom data types used by the external functions of the asset, and allows the user to select any additional data types that are part of the API for the asset; paragraph 0130].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention of Sadhu to include the features of Carlson. One of ordinary skill in the art would have been motivated to make the combination because this provides generalized interfaces to diverse repositories, and to generate software assets in a normalized form that complies with a data description language

[paragraph 0014 of Carlson]. Sadhu and Carlson do not teach an artifact type comprising a defect.

However, Chulani teaches an artifact type comprising a defect [four types of defect artifacts; Section 3.1].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to further modify the invention of Sadhu and Carlson to include the features of Chulani. One of ordinary skill in the art would have been motivated to make the combination because this allows defects to be introduced in several activities of the software development life cycle and classified based on their origin [Section 3.1 of Chulani].

20. As to claim 2, Sadhu teaches the first application is a defect tracking application [paragraphs 0063 – 0064 of Carlson]; and wherein the second application is a source control application [paragraph 0060 of Carlson].

21. As to claim 3, Sadhu teaches the first application is a source control application [paragraph 0060 of Carlson]; and wherein the second application is a defect tracking application [paragraphs 0063 – 0064 of Carlson].

22. As to claim 4, Sadhu teaches a linking component that links the reference with the corresponding artifact of the first application [col. 5, lines 32 – 58; col. 9, lines 5 – 13 and lines 50 – 57].

23. As to claim 5, Sadhu as modified teaches the linking component is an artifact identifier held by the artifact consumer that points to an artifact [paragraph 0118 of Carlson].

24. As to claim 6, Sadhu as modified teaches the link is a binary link [paragraph 0097 of Carlson].

25. As to claim 7, Sadhu teaches at least one of the provider and the consumer is a tool or service [col. 7, lines 4 – 24].

26. As to claim 8, Sadhu teach the artifact provider registers an artifact type for each artifact it provides, and registers a corresponding link type that each artifact can host [col. 18, lines 15 – 20].

27. As to claim 9, Sadhu teaches a generic artifact provider (GAP) that interfaces to a tool to facilitate storing and exposing both artifacts and artifact links [col. 3, lines 56 – 65].

28. As to claim 10, Sadhu teaches a GAP adapter that provides an interface between the GAP and a non-integrated application [col. 3, lines 56 – 65].

29. As to claim 11, Sadhu teaches a cache that stores the artifacts and associated artifact links [database 32; col. 5, lines 32 – 47 and col. 8, lines 6 – 23].
30. As to claim 12, Sadhu teaches a user interface that facilitates presenting inter-artifact references [col. 5, lines 11 – 19].
31. As to claim 13, Sadhu teaches a computer readable storage medium having stored thereon computer executable instructions for carrying out the system of claim 1 [col. 3, lines 40 – 48].
32. As to claim 17, Sadhu as modified teaches the link is an artifact identifier that is an immutable and uniquely constructed key [paragraph 0118 and 0132 of Carlson].
33. As to claim 18, Sadhu teaches a link manager that manages a cache by updating and purging cache contents [col. 12, lines 48 – 59].
34. As to claim 19, Sadhu teaches the artifact provider and artifact consumer are at least one of loosely coupled and tightly coupled [col. 12, lines 36 – 47].
35. As to claim 20, Sadhu teaches a classifier that makes an inference based on parameters related to at least one of the artifact consumer, artifact provider, and non-integrated applications [col. 11, lines 41 – 49].

36. As to claim 21, Sadhu as modified teaches the artifact provider creates and reveals a URI [paragraph 0132 of Carlson] for at least one of loosely-coupled server-based interactions, loosely-coupled clients, caching, and tightly-coupled interactions that support artifact-specific functions by contract with a caller [col. 12, lines 36 – 47 of Sadhu].

CONTACT INFORMATION

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to LI B. ZHEN whose telephone number is (571)272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sub Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194